Mahidol University *Wisdom of the Land*

# Chapter 6

# Image Restoration in Spatial domain

# Image Restoration

**Noisy image**
***f*(x,y)**

→

**Better quality image**
***g*(x,y)**

**Cause:**

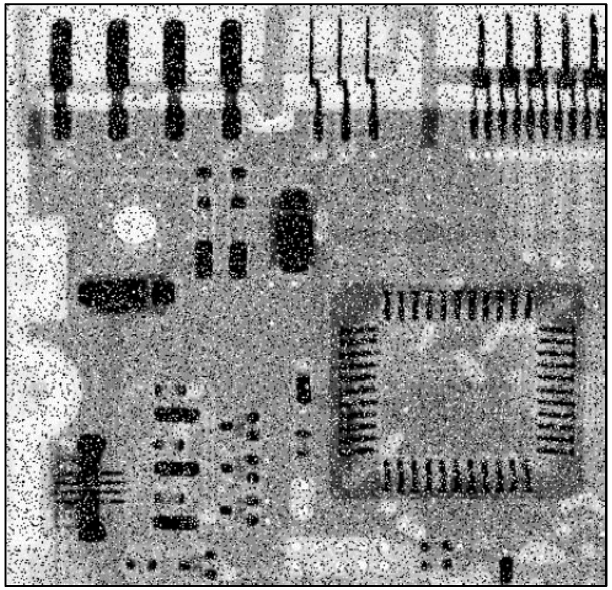
a) Noise can be added to images during image acquisition (faulty CCD elements);
b) during image transmission (channel interference).

**Restoration:**

a) Spatial domain
b) Frequency domain

# What is Image Restoration?

- Restoration is to attempt to reconstruct or recover an image that has been degraded by using a priori knowledge of the degradation.
- The restoration approach involves the modeling of the degradation and applying the reverse process to recover the original image.



Noisy image

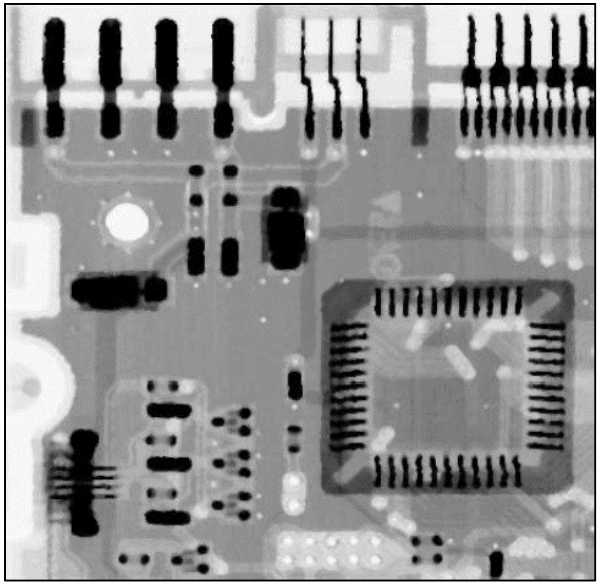


Image obtained using a median filter

# Image Restoration VS. Image Enhancement

- **Image restoration and image enhancement**
  - The goal of both approaches is to improve image quality.

- **Image Restoration** is to restore a degraded image back to the original image, based on formulate a criterion of goodness that will yield an optimal estimate of the desired result (objective process).

- Techniques include noise removal.

- **Image Enhancement** is to manipulate an image that it is suitable for a specific application, based on satisfy human visual system (subjective process).

- Techniques include the gray-level transformation operations.

# Image Restoration

**Noisy image**
***f*(x,y)**

**Better quality image**
***g*(x,y)**

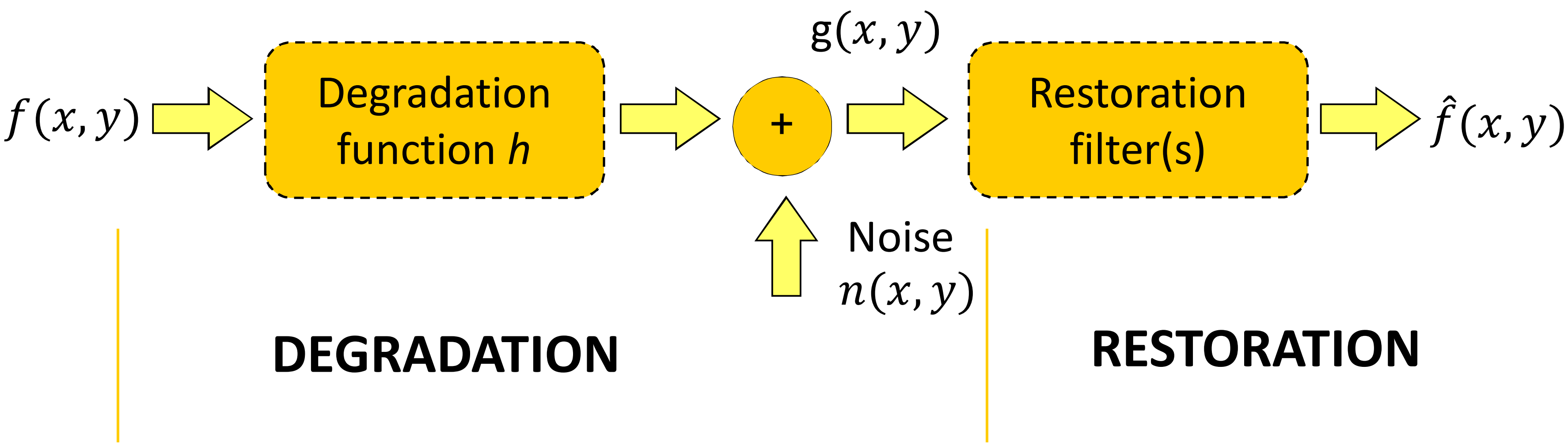
**Noise Models:**

a) Gaussian
b) Rayleigh
c) Erlang (Gamma)
d) Exponential
e) Uniform
f) Impulse (salt and Pepper)

**Noise Removal:**

a) Arithmetic mean filter
b) Geometric mean filter
c) Harmonic mean filter
d) Contra-harmonic mean filter
e) Median filter
f) Max, Min filters
g) Mid-point filter
h) Alpha-trimmed mean filter
i) Adaptive local noise reduction filter
j) Adaptive mean filter

# A Model of the Image Degradation/Restoration Process




- Where
  - $f(x,y)$ : an input image
  - $g(x,y)$ : a degraded image
  - $h(x,y)$ : the degradation function
  - $n(x,y)$ : the additive noise
  - $\hat{f}(x,y)$ : an estimate of the original image

The more we know about $h$ and $n$, the closer $\hat{f}(x,y)$ will be to $f(x,y)$

# A Model of the Image Degradation/Restoration Process

- If degradation function $h$ is a linear, position-invariant process, the degraded image in the spatial domain representation can be modeled by :

$$g(x, y) = h(x, y) * f(x, y) + n(x, y)$$

- Where : $h(x, y)$ is a system that causes image distortion and $h(x, y)$ is noise.
- We will assume first that $h$ is an identity operator and all degradation is due to additive noise.
- In almost all considerations we will assume that the noise is uncorrelated with the image, which means that there is no correlation between the pixel values of the image and the noise components.

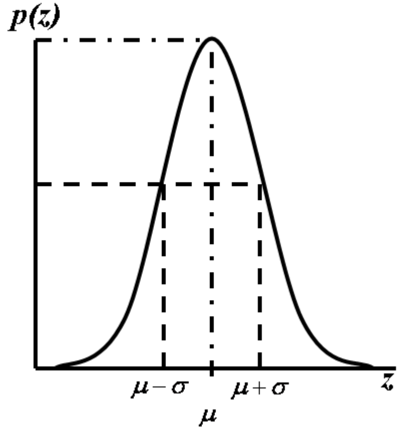
# Noise Models

- Most types of noise are modeled as known probability density functions (PDF).
- Noise model is decided based on understanding of the physics of the sources of noise.
  - **Gaussian noise**
  - **Rayleigh noise**
  - **Erlang (Gamma) noise**
  - **Exponential noise**
  - **Uniform noise**
  - **Impulse (Salt and Pepper) noise**
- Parameters can be estimated based on histogram on small flat area of an image.

- **Gaussian Noise** : Gaussian noise models are frequently used in practice. Because this type of noise model is easily tractable in both spatial and frequency domains.
- The PDF of the Gaussian random variable, $z$ is given by :

$$\boxed{p(z) = \frac{1}{2\pi\sigma} e^{-(z-\mu)^2/2\sigma^2}}$$

- Where :
  - $z$ : gray-level
  - $\mu$ : the mean of average value of $z$
  - $\sigma$ : its standard deviation
  - $\sigma^2$ : variance of $z$

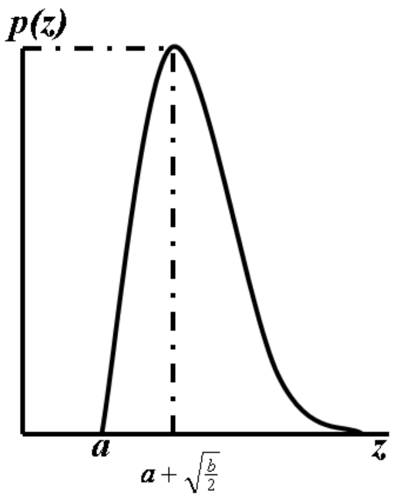


- **Rayleigh Noise** : The PDF of Rayleigh noise is given by :

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for z < a} \end{cases}$$

- The mean of the density is : $u = a + \sqrt{\pi b / 4}$

- The variance of the density is : $\sigma^2 = \frac{b(4-\pi)}{4}$

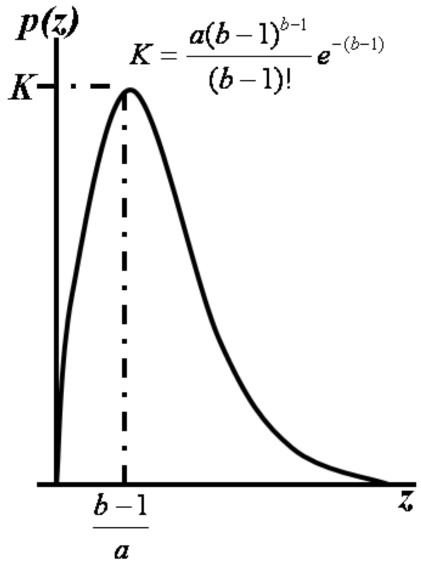



- Because of its skewed distribution, it can be useful for approximating the distribution of the images characterized by skewed histograms.

# Erlang (Gamma) Noise

- **Erlang (Gamma) Noise** : The PDF of Gamma noise is given by :

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!}(z-a)e^{-az} & \text{for } z \ge 0 \\ 0 & \text{for z} < 0 \end{cases}$$




- The mean of the density is : $u = \dfrac{b}{a}$

- The variance of the density is : $\sigma^2 = \dfrac{b}{a^2}$

- When the denominator is the gamma function, the PDF describes the gamma distribution.

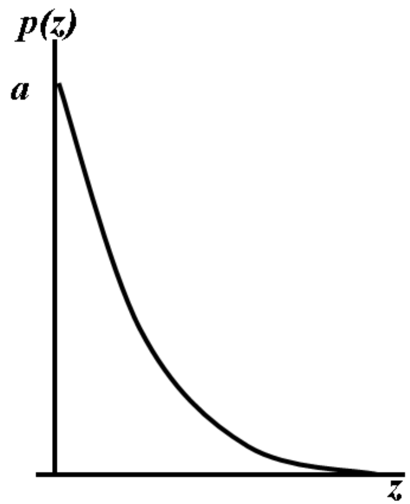
# Exponential Noise

- **Exponential Noise** : The PDF of Exponential noise is given by :

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

- The mean of the density is : $u = \dfrac{1}{a}$

- The variance of the density is : $\sigma^2 = \dfrac{1}{a^2}$




- The PDF of the Exponential noise is the special case of the Gamma PDF, where b=1.

# Uniform Noise

- **Uniform Noise** : The PDF of Uniform noise is given by :

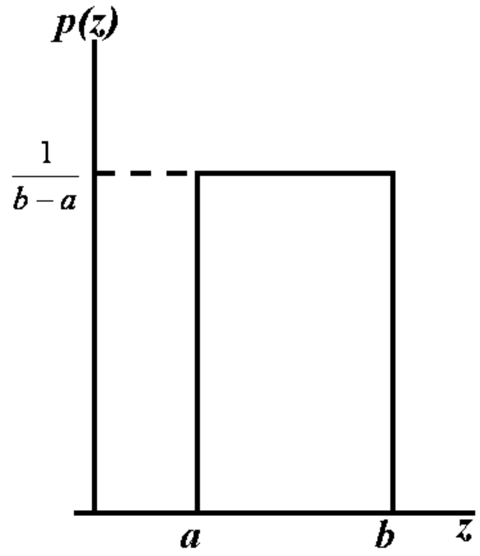
$$p(z) = \begin{cases} \dfrac{1}{\text{b - a}} & \text{for a} \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

- The mean of the density is :

$$u = \frac{a+b}{2}$$

- The variance of the density is :

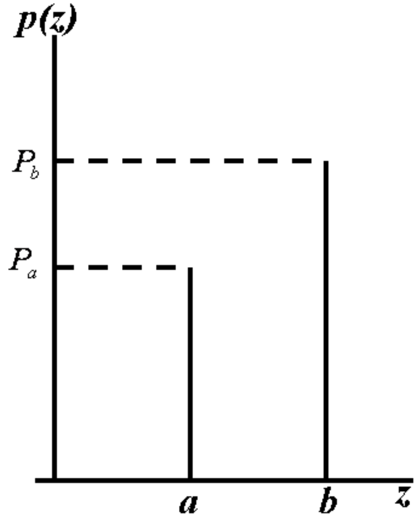
$$\sigma^2 = \frac{(b-a)^2}{12}$$



# Impulse (salt-and-pepper) Noise

- Impulse (salt-and-pepper) Noise : The PDF of (bipolar) impulse noise is given by :

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$




- If $b > a$, ➔ the gray-level $b$ will appear as a light dot in the image.

  ➔ the gray-level $a$ will appear like a dark dot.
- If $P_a = P_b = 0$ ➔ the impulse noise is called unipolar.
- If $P_a = P_b \neq 0$ ➔ impulse noise value will resemble salt-and-pepper (or shot-and-spike) granules randomly distributed over the image.

# Noise Models

The preceding PDFs provide useful tools for modeling a broad range of noise corruption situations found in practice.

- **Gaussian noise** : electronic circuit noise and sensor noise due to poor illumination and/or high temperature.
- **Rayleigh noise** : characterizing noise phenomena in range imaging.
- **Exponential** and **Gamma noise** : application in laser imaging.
- **Uniform noise** : useful as the basis for numerous random number generators that are used in simulations.
- **Impulse noise** : quick transients, such as faulty switching, take place during imaging.

# Estimation of Noise Parameters

- The simplest way to use the data from the image strips is for calculating the mean and variance of the gray levels.
- If $S$ is a strip (sub-image),

$$\boxed{\mu = \sum_{z_i \in S} z_i p(z_i)}$$

$$\boxed{\sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)}$$

- Where :
  - $Z_i$ : the gray-level values of the pixels in $S$
  - $p(z_i)$ : the corresponding normalized histogram values.
- The shape of the histogram identifies the closest PDF match.
- If the shape is approximately Gaussian, then mean and variance is all we need because the Gaussian PDF is completely specified by these two parameters.

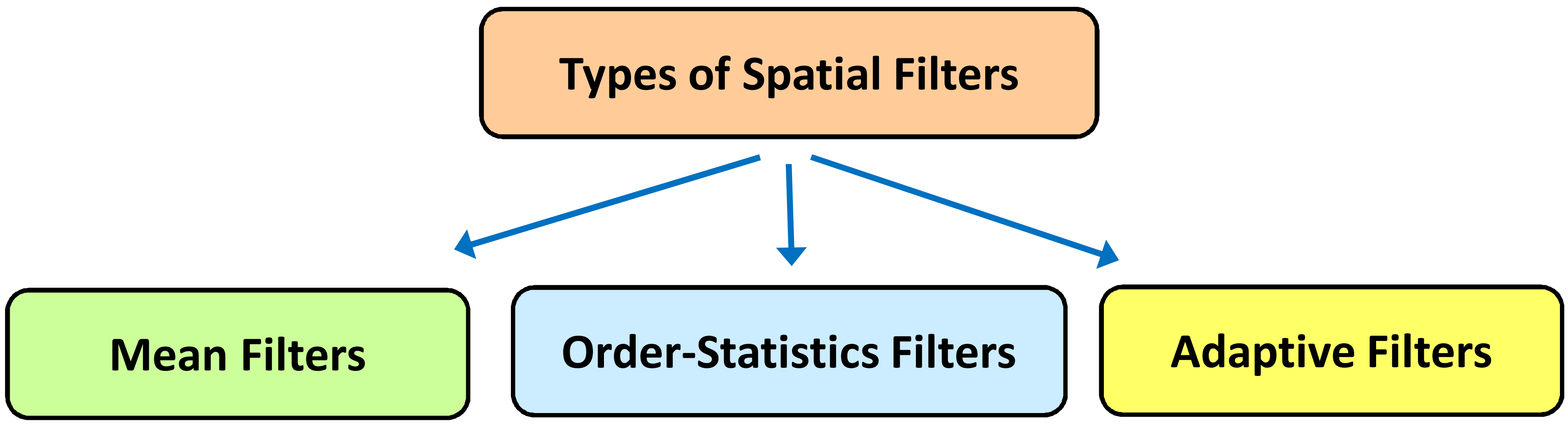
# Estimation of Noise Parameters

- When the only degradation present in an image is noise.

$$g(x,y) = f(x,y) + \eta(x,y)$$

- The noise terms are unknown, so subtracting then is not a realistic option.
- Spatial filtering is the method of choice in situations when only additive noise is present.

# Restoration methods



# Mean Filters

- **Mean Filters (Linear Filters)** : There are four main types of noise reduction that can be used for image restoration/enhancement.
  - Arithmetic mean filter
  - Geometric mean filter
  - Harmonic mean filter
  - Contra-harmonic mean filter

**Arithmetic mean filter (Average filter)**

- Let $S_{xy}$ be the coordinates in a sub-image window of size $mxn$ centered at point $(x, y)$. The value of the restored image at any point $(x, y)$ is given by :

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

- This operation can be implemented by a convolution mask in which all its components have a value $\frac{1}{mn}$
- Effects : Local variations in the image is smoothed and noise is reduced as a result of blurring.

**Geometric mean filter**

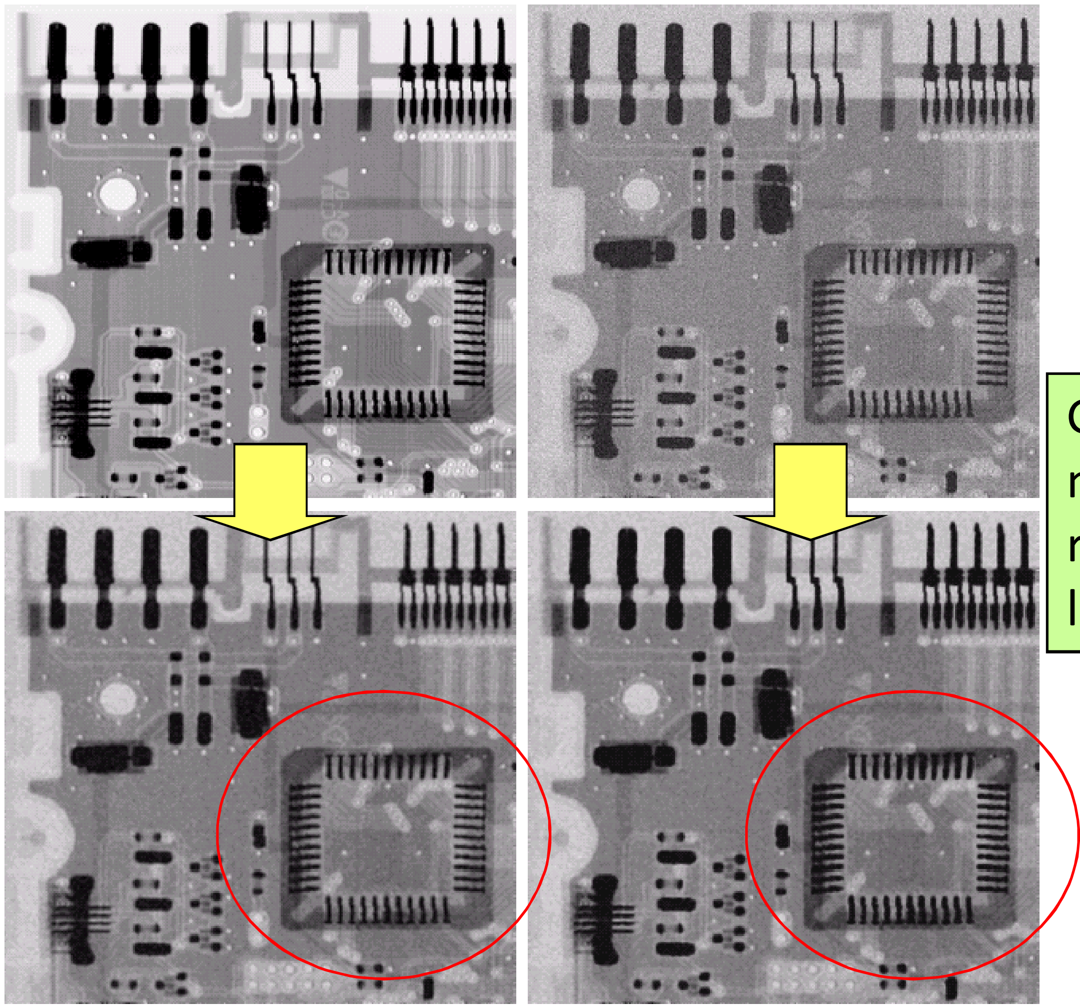
- The value of the restored image at any point $(x, y)$ is given by :

$$\hat{f}(x,y) = \left( \prod_{(s,t)\in S_{xy}} g(s,t) \right)^{\frac{1}{mn}}$$

- In this filter, each pixel is given by the product of the pixels in the sub-image windows, raised to the power of $\frac{1}{mn}$
- Effects : Achieves more smoothing, but looses less image details.

# Arithmetic vs. Geometric Mean Filters

Gaussian Noise



Arithmetic mean noise is reduced at the expense of blurring.

Geometric mean noise is also removed, but less blurring.

**Harmonic mean filter**

- The value of the restored image at any point $(x, y)$ is given by :

$$\hat{f}(x,y) = \frac{mn}{\displaystyle\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

- Harmonic mean filter works well with the salt noise but fails for the pepper noise.
- Effects : It works well with other types of noise as well, such as Gaussian noise.

**Contra-harmonic mean filter**

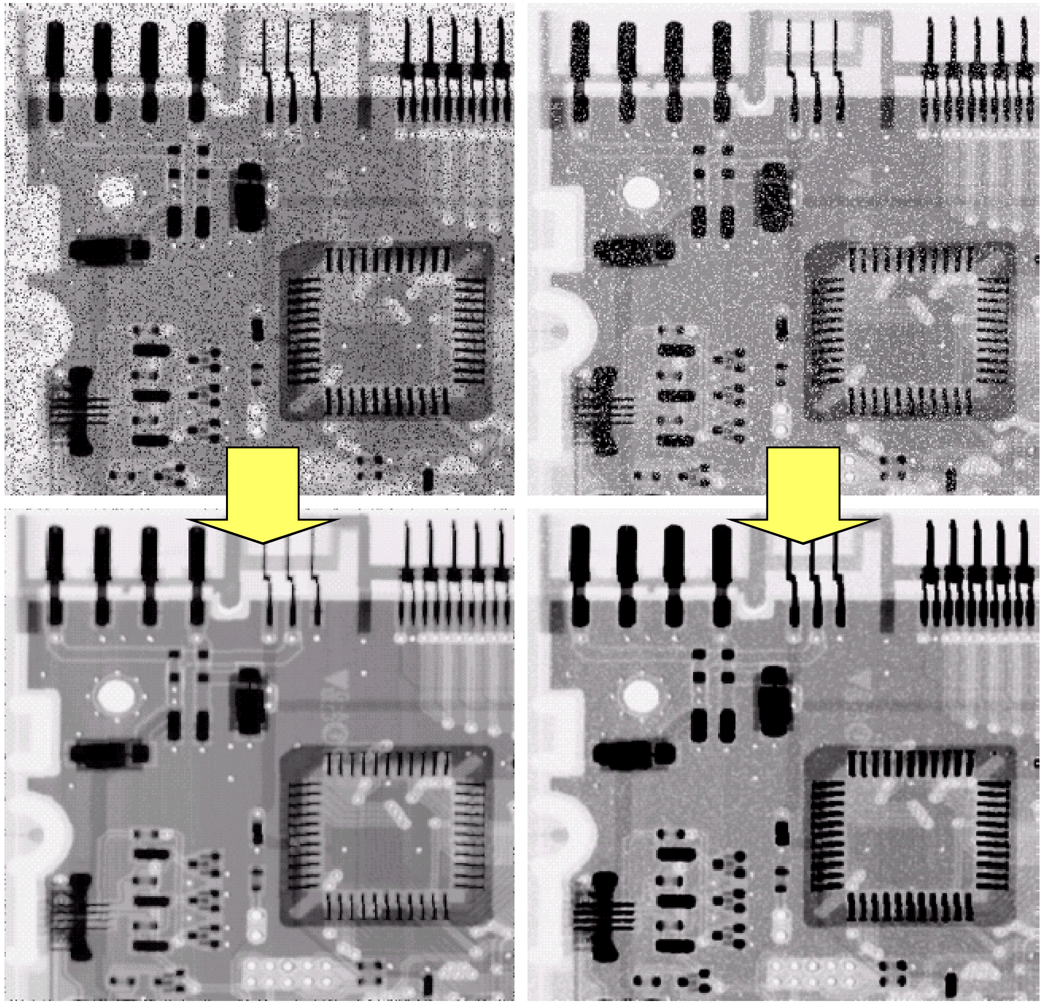
- The value of the restored image at any point $(x, y)$ is given by :

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

- Effects : well suited for Salt and Pepper noise :
- For negative values of $Q$, it eliminates the Salt noise and for positive values of $Q$, it eliminates the Pepper noise.
- The filter becomes the arithmetic mean filter for $Q = 0$, and harmonic mean filter for $Q$=-1

# Example : Contra-harmonic Filter

Slat and Pepper noise



$Q > 0$ filter did a good job in cleaning the background at the expense of some blur in dark areas.

the opposite is true for Q < 0

# Contra-harmonic Filters: Incorrect Use Example

Be careful of wrong sign in contra-harmonic filtering!

Image corrupted by Pepper noise with prob. = 0.1

Image corrupted by Salt noise with prob. = 0.1

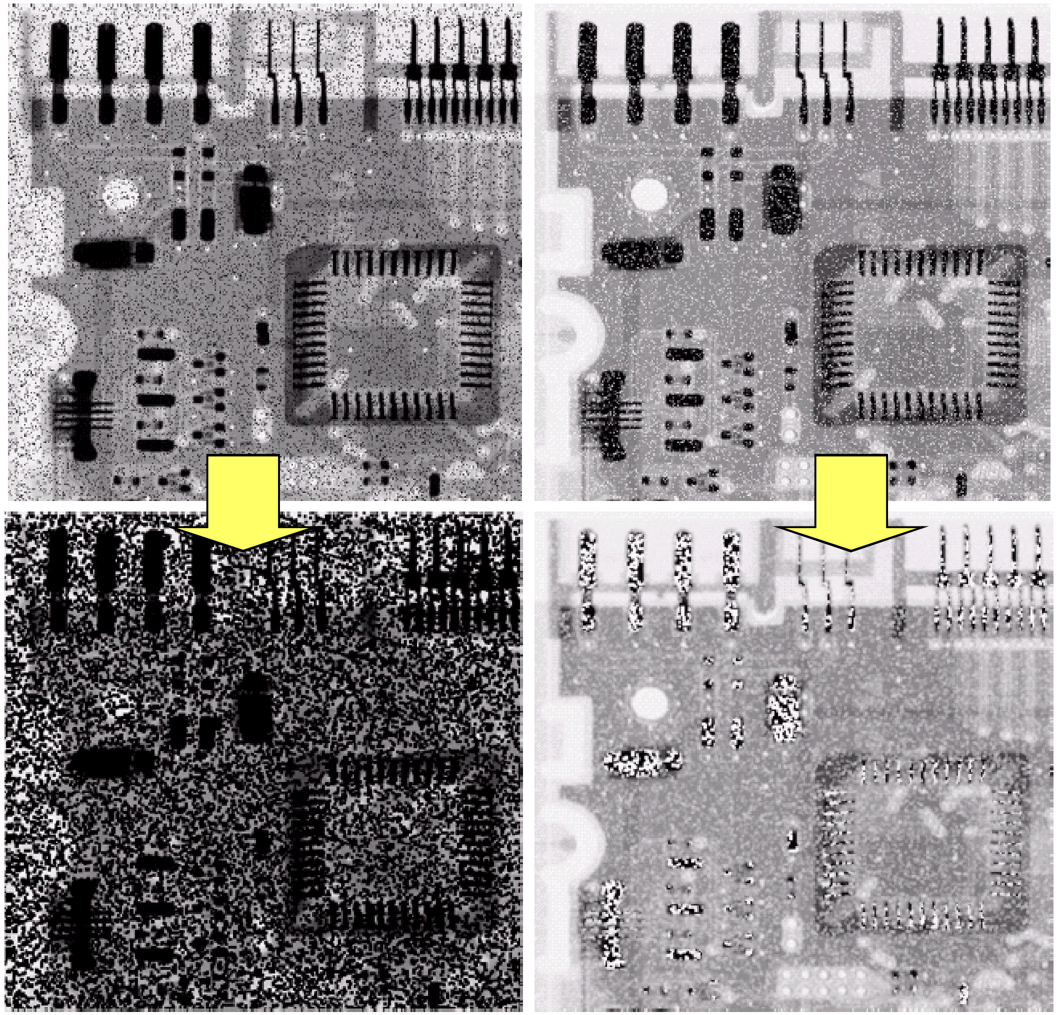


Image obtained using a 3x3 contra-harmonic mean filter With $Q$=-1.5

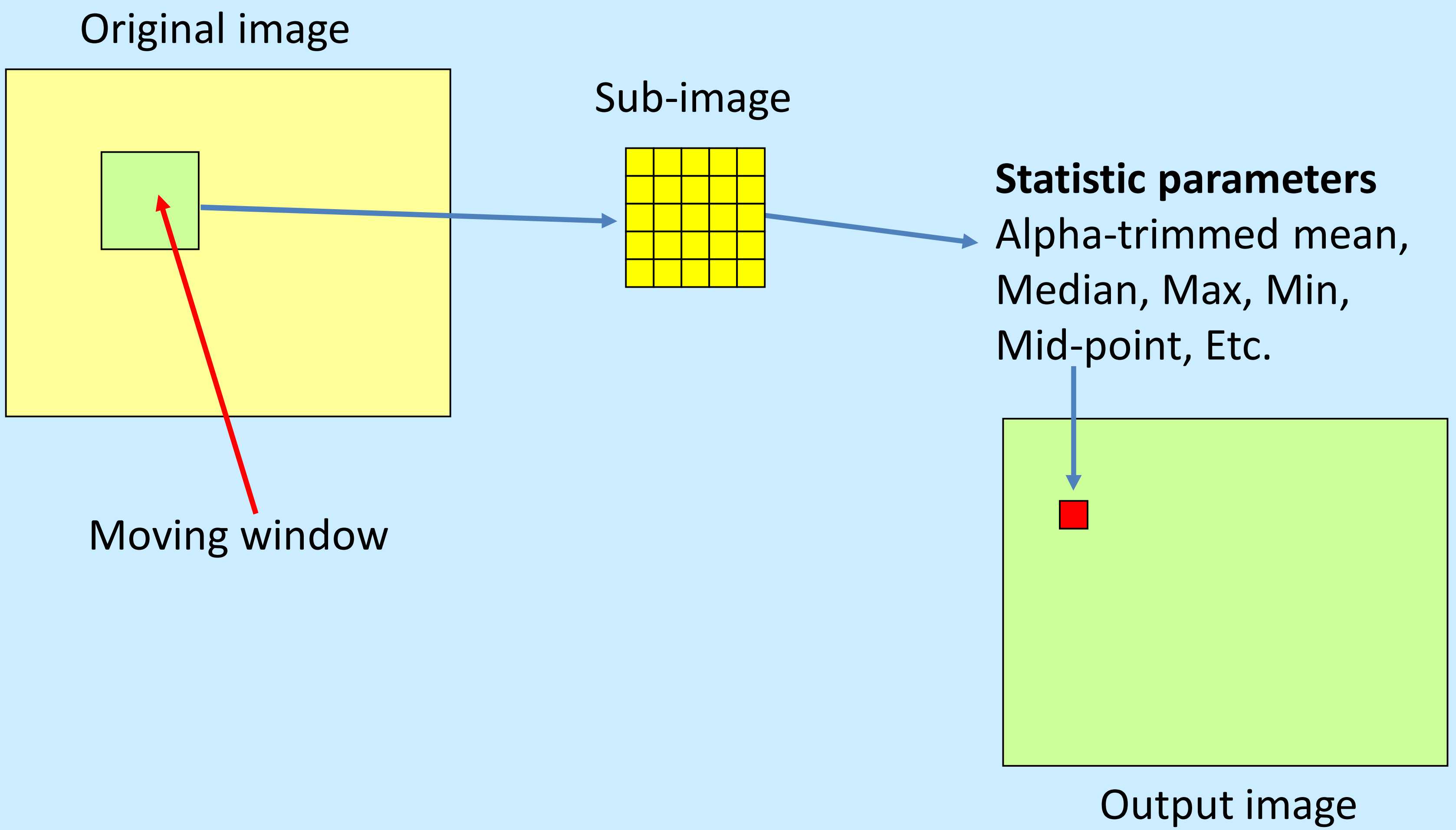
Image obtained using a 3x3 contra-harmonic mean filter With $Q$=1.5

# Order-Statistics Filters

- **Order-Statistics Filters (Non-Linear Filters)** are spatial filters whose response is based on ranking the pixels contained in the image area surrounded by the filter.
  - Median filter
  - Max and Min filters
  - Mid-point filter
  - Alpha-trimmed mean filter

# Order-Statistics Filters

- Based on ranking the input pixels contained in a local window and selecting one of them as the output pixel.



**Median filter**

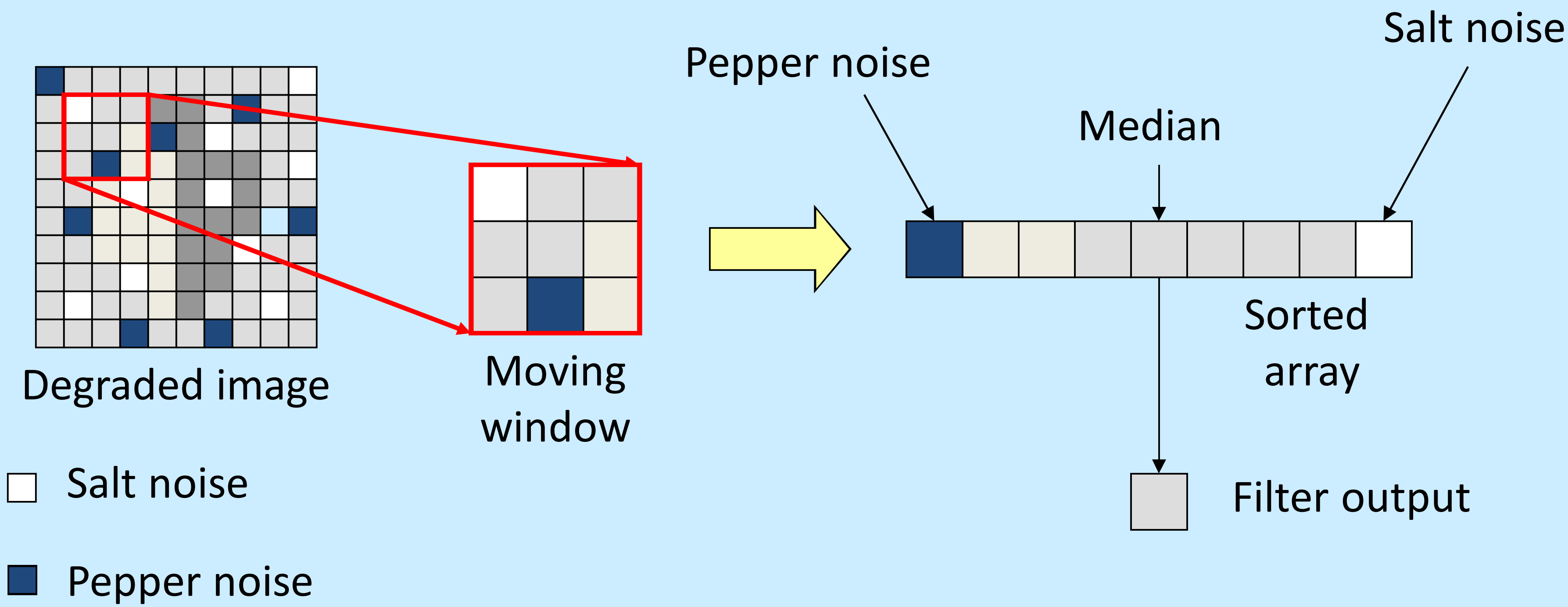
- Let $S_{xy}$ be the coordinates in a sub-image window of size $m \times n$ centered at point $(x, y)$. The value of the restored image at any point $(x, y)$ is given by :

$$\boxed{\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{g(s,t)\}}$$

- The value of a pixel in is replaced by the median of the gray-level in the neighborhood characterized by $S_{xy}$ sub-image.
- Provides less blurring than the other linear smoothing filters.
- Median filters are very effective in bipolar or unipolar impulse (Salt and Pepper) noise.

**Median filter : How it works?**

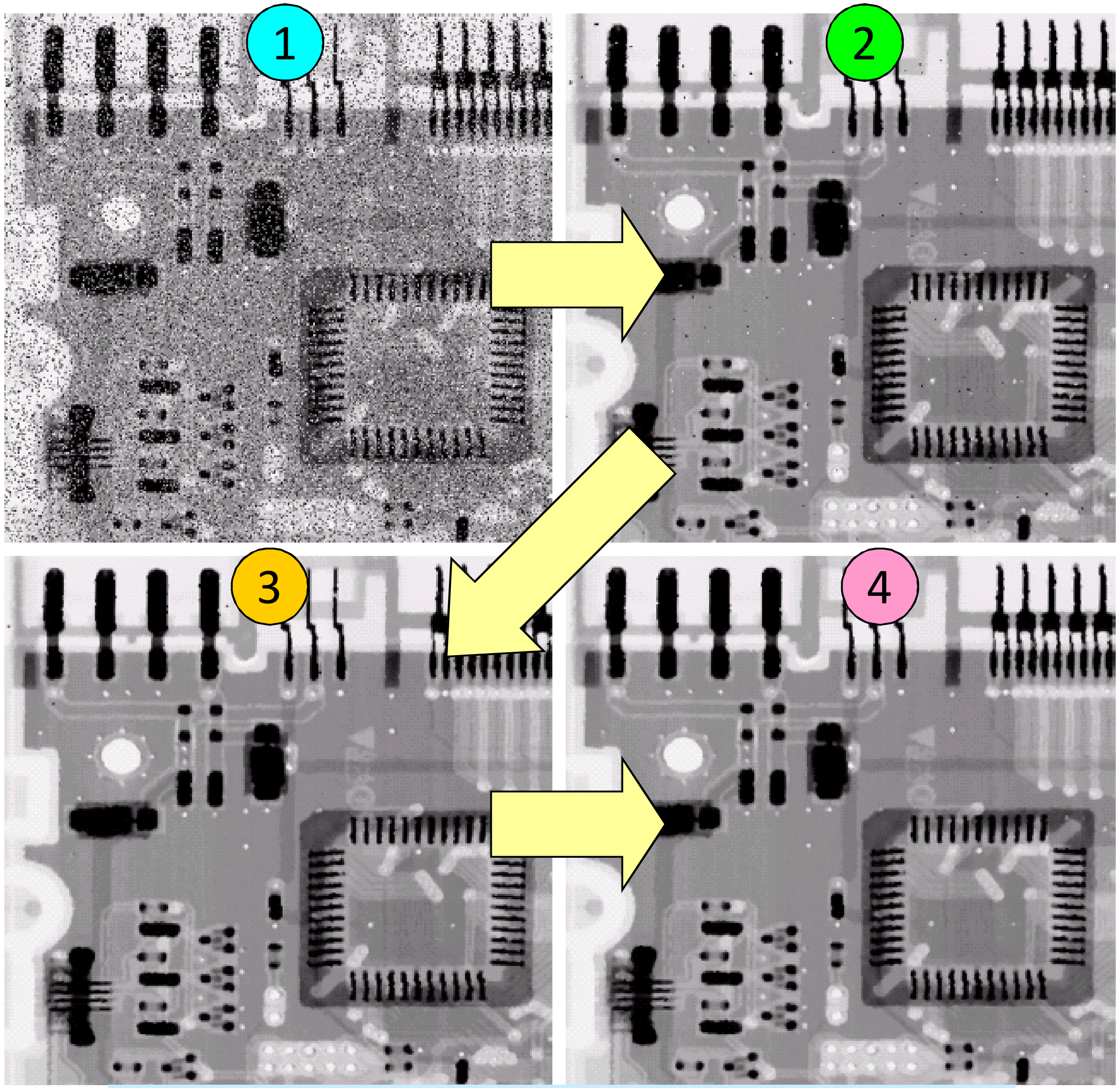
- A median filter is good for removing impulse (Salt and Pepper ) noise.




- Therefore, it is rare that the noise pixel will be a median value.

# Example : Median Filter

Image corrupted by Salt and Pepper noise with $p_a=p_b= 0.1$




Images obtained using a 3x3 median filter

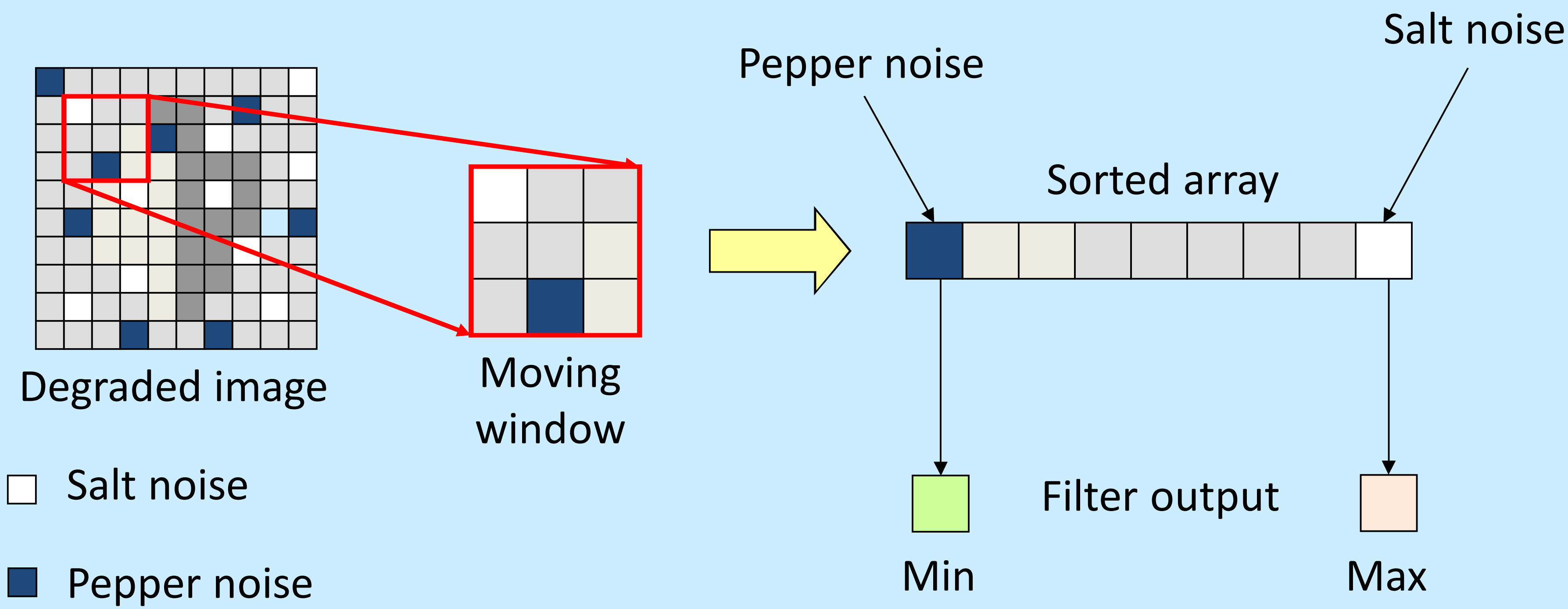
# Order-Statistics Filters

**Max and Min filters**

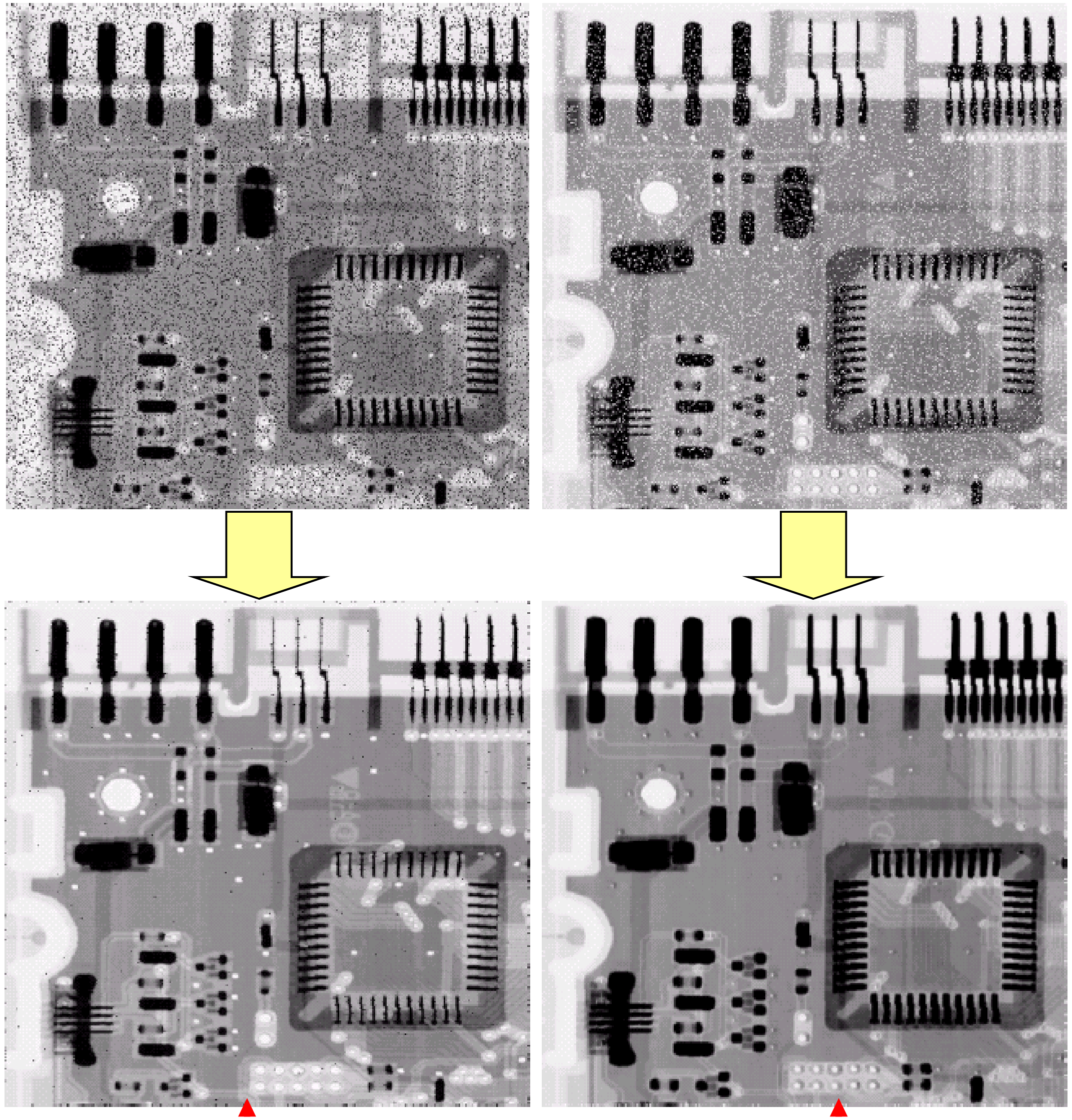
- The value of the restored image at any point $(x, y)$ is given by :
  - Max filter $$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$ ⇒ Reduce "dark" noise (Pepper noise)
  - Min filter $$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$ ⇒ Reduce "bright" noise (Salt noise)
- Max filter is useful for finding the brightest points in an image. It also reduces Pepper noise.
- Min filter is useful for finding the darkest points in an image. Thus, it can reduces Salt noise.

## Max and Min filters : How it works?

- Max filter reduces Pepper noise.
- Min filter reduces Salt noise.



# Example : Max and Min Filters



result of filtering the image corrupted by Pepper noise with the Max filter

result of filtering the image corrupted by Salt noise with the Min filter

**Mid-point filter**

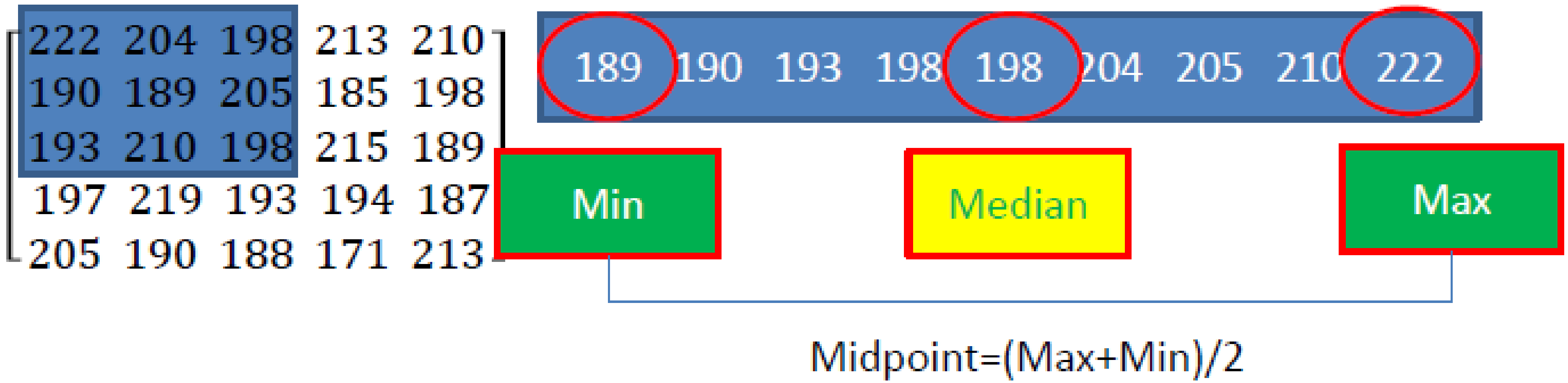
- The value of the restored image at any point $(x, y)$ is given by :

$$\hat{f}(x,y) = \frac{1}{2}\left( \max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\} \right)$$

- Computes the midpoint between the maximum and minimum values of intensities.
- This filter combines the order statistics and averaging.
- Works best for randomly distributed noise, like Gaussian and uniform noise.

**Mid-point filter : How it works?**

- Computes the mid-point between the maximum and minimum values of intensities.



# Order-Statistics Filters

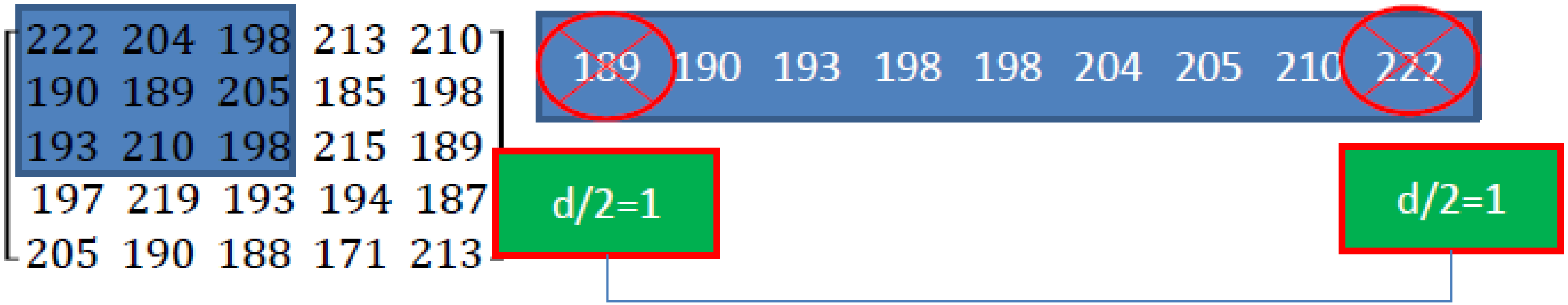
**Alpha-trimmed mean filter**

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

- Suppose we remove $\frac{d}{2}$ lowest and $\frac{d}{2}$ highest samples from the local input set and average the remaining samples.
- Combines order statistics and averaging.
- Works well for combination of noises, like Gaussian and Salt & Pepper noises.

**Alpha-trimmed mean filter : How it works?**

- Computes the order-statistics and averaging.




Alpha-trimmed mean=(190+193+198+204+205+210)/7

# Example : Alpha-trimmed mean filter

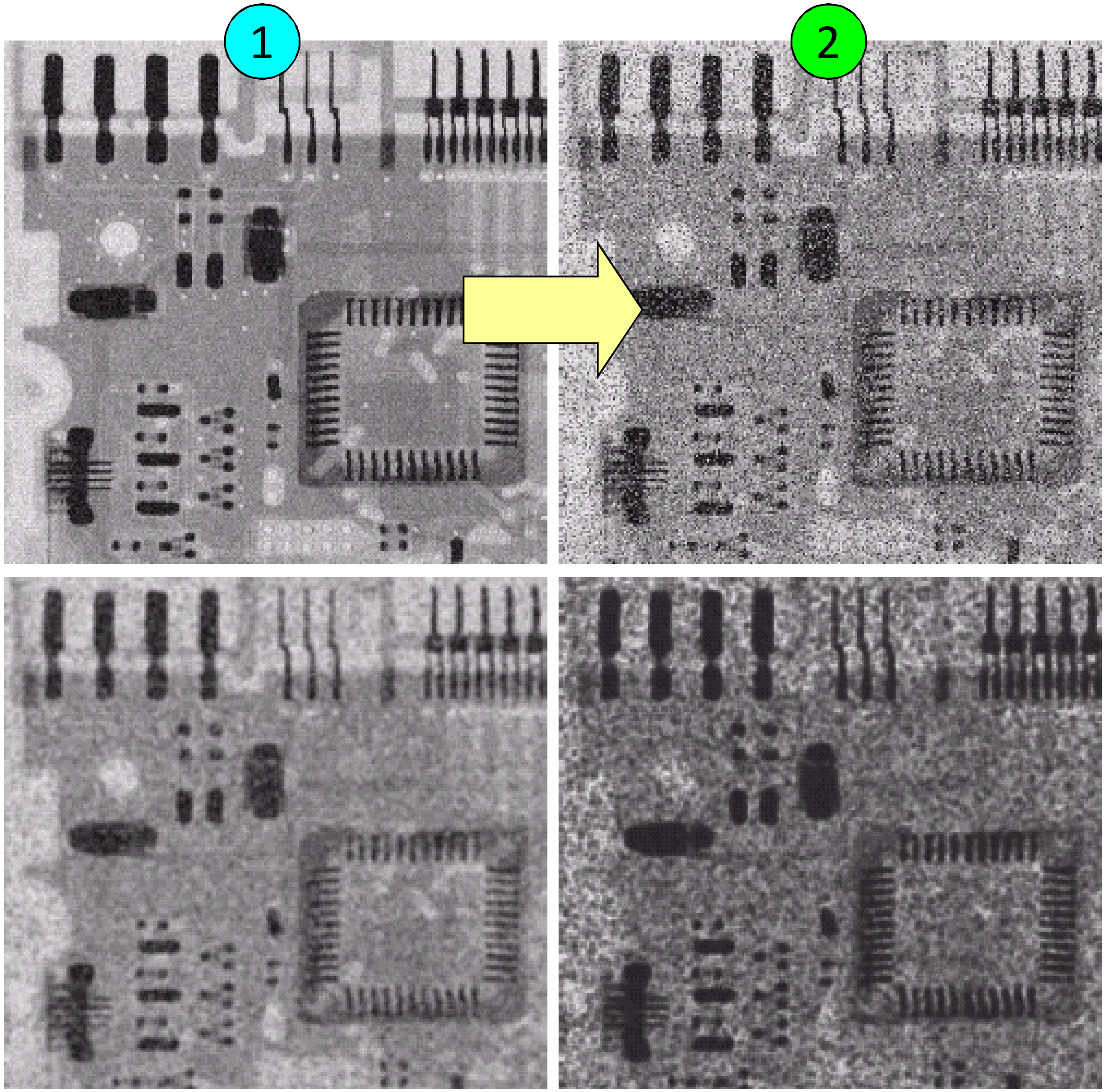



Image corrupted by additive uniform noise

Image additionally corrupted by additive salt-and-pepper noise

Image 2 obtained using a 5x5 arithmetic mean filter

Image 2 obtained using a 5x5 geometric mean filter

# Example : Alpha-trimmed mean filter

Image corrupted by additive uniform noise

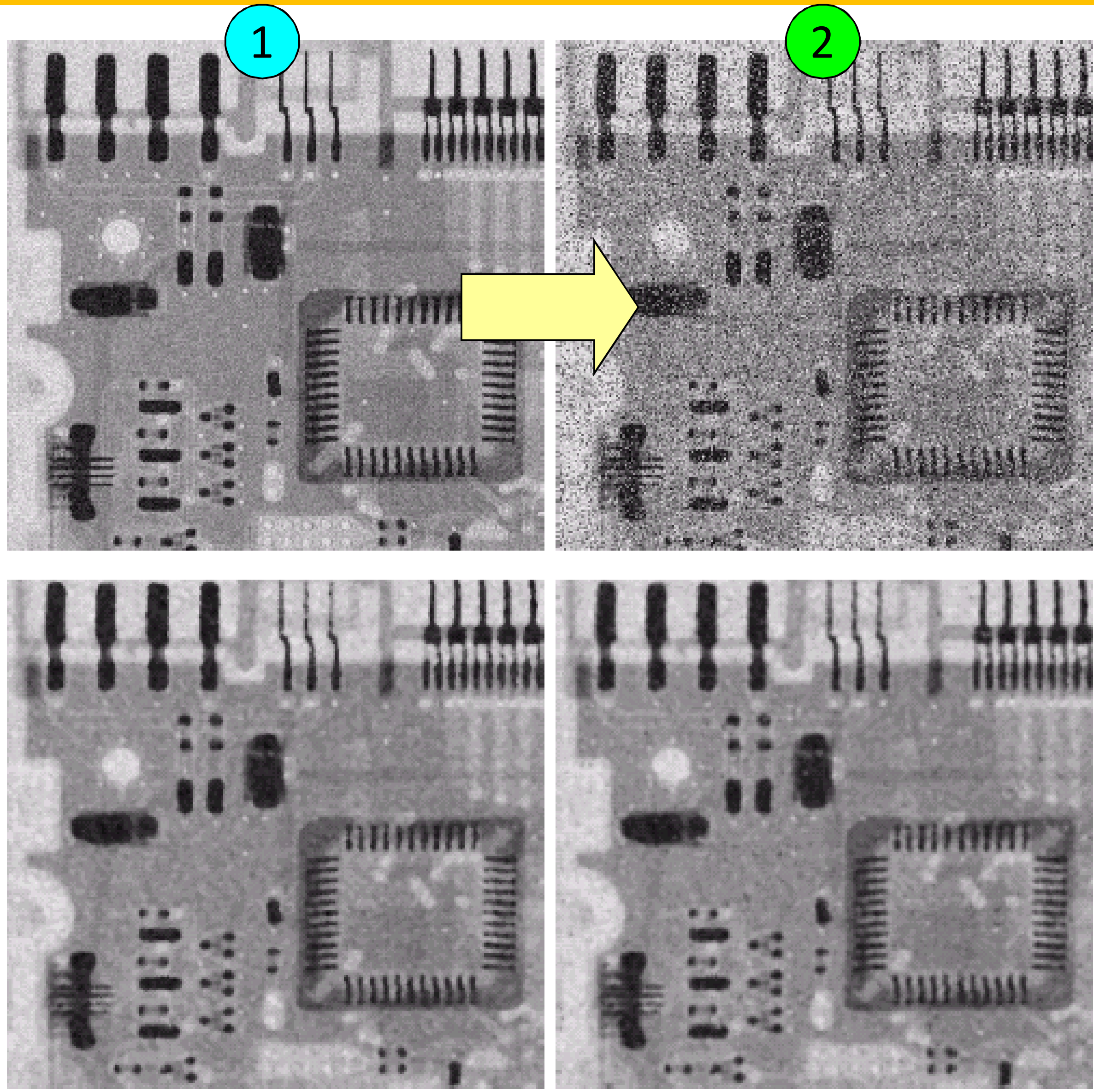



Image additionally corrupted by additive salt-and-pepper noise

Image 2 obtained using a 5x5 median filter

Image 2 obtained using a 5x5 alpha-trimmed mean filter with $d = 5$

# Restoration methods

- **Adaptive Filters** : The behavior of adaptive filters changes based on statistical characteristics of the image inside the filter region.
  - Adaptive local noise reduction filter
  - Adaptive median filter

# Adaptive Filters

General concept :

- Filter behavior depends on statistical characteristics of local areas inside m x n moving window.
- More complex but superior performance compared with "fixed" filters.

- Statistical characteristics :
  - Local mean :

$$\boxed{m_L = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)}$$

  - Local variance :

$$\boxed{\sigma_L^2 = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} (g(s,t) - m_L)^2}$$

  - Noise variance :

$$\boxed{\sigma_\eta^2}$$

# Adaptive Filters

**Adaptive local noise reduction filter**

- Mean and variance are the simplest statistical measures of a random noise.
  - Mean gives the measure of the average gray-level in a local region;
  - Variance gives the measure of the average contrast in the region.
- Consider a filter operating in a local region, $S_{xy}$, where the response of the filter at any point $(x, y)$ depends on :

  a) $\mathrm{g}(x, y)$, the value of the noisy image at $(x, y)$

  b) $\sigma_\eta^2$ , the additive noise variance,

  c) $m_L$ , local mean of pixels in $S_x$.

  d) $\sigma_L^2$ , the local variance in $S_{xy}$.

## Adaptive local noise reduction filter

- Purpose : want to preserve edges

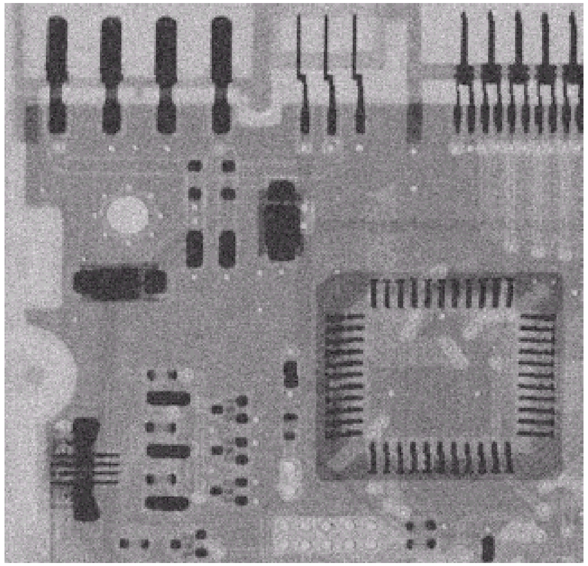
1. If $\sigma_h^2$ is zero, → No noise
   the filter should return $g(x,y)$ because $g(x,y) = f(x,y)$
2. If $\sigma_L^2$ is high relative to $\sigma_h^2$, → Edges (should be preserved),
   the filter should return the value close to $g(x,y)$
3. If $\sigma_L^2 = \sigma_h^2$, → Areas inside objects
   the filter should return the arithmetic mean value $m_L$

- Formula :

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left(g(x,y) - m_L\right)$$

# Example : Adaptive local noise reduction filter

Image corrupted by additive Gaussian noise with zero mean and $\sigma_\eta^2 = 0.001$



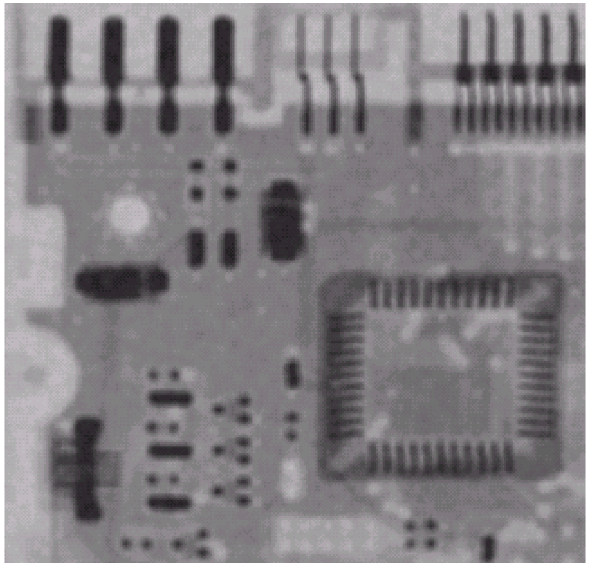
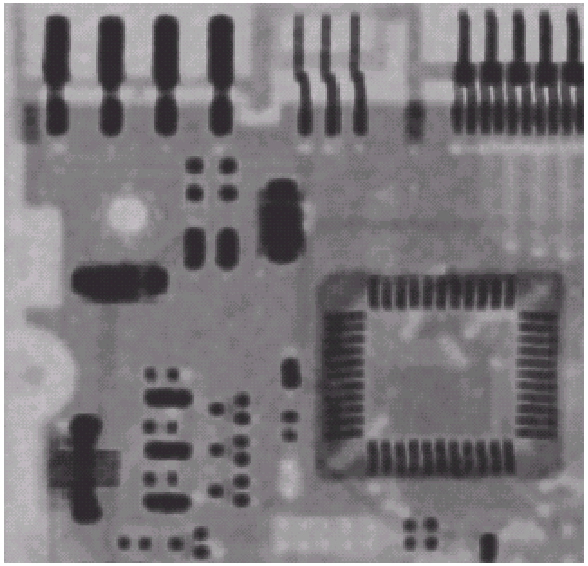


Image obtained using a 7x7 arithmetic mean filter

Image obtained using a 7x7 geometric mean filter



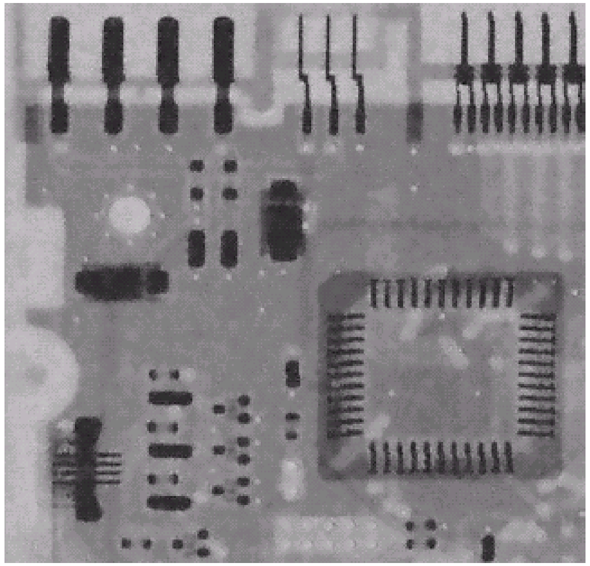


Image obtained using a 7x7 adaptive noise reduction filter

# Adaptive Filters

**Adaptive median filter** : can filter impulse noise with very high probabilities ($p_a$>=$p_b$ >=0.2). Additionally smooth the non-impulse noise which is not the feature of a traditional median filter.

- The filter uses the following parameters in the neighborhood of $S_{xy}$ :

  $z_{min}$ = minimum gray level value in $S_{xy}$
  $z_{max}$ = maximum gray level value in $S_{xy}$
  $z_{median}$ = median of gray levels in $S_{xy}$
  $z_{xy}$ = gray level value at pixel $(x,y)$
  $S_{max}$ = maximum allowed size of $S_{xy}$

- Note that unlike the other filters the size of $S_{xy}$ increases during the filtering operation.
- Changing size of the filter mask does not change the fact that the output of the filter is still a single value centering the mask.

# Adaptive Median Filter : How it works

Level *A*: A1= $z_{median} - z_{min}$

A2= $z_{median} - z_{max}$

If A1 > 0 and A2 < 0, goto level B

Else → Window is not big enough

increase window size

If window size <= $S_{max}$ repeat level A

Else return $z_{xy}$

Determine whether $z_{median}$ is an impulse or not

Level *B*:

→ $z_{median}$ is not an impulse

B1= $z_{xy} - z_{min}$

B2= $z_{xy} - z_{max}$

If B1 > 0 and B2 < 0,

return $z_{xy}$ → to preserve original details

Else

return $z_{median}$ → to remove impulse

Determine whether $z_{xy}$ is an impulse or not

→ $z_{xy}$ is not an impulse

# Example : Adaptive Median Filter

More small details are preserved

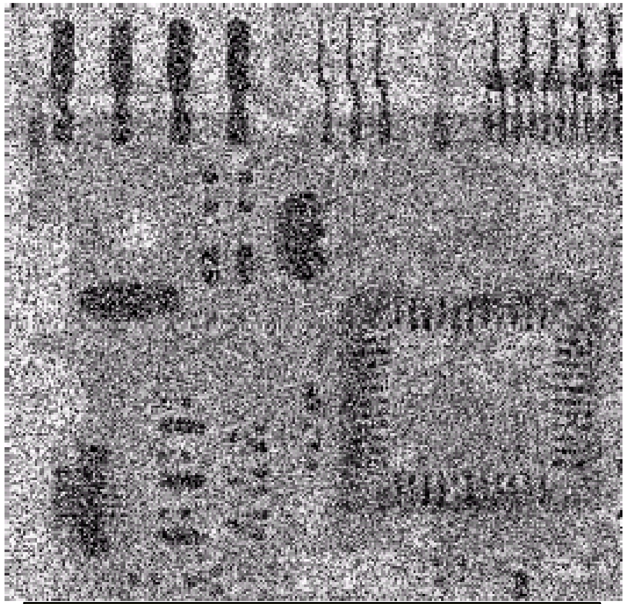


Image corrupted by salt-and-pepper noise with $p_a$=$p_b$= 0.25

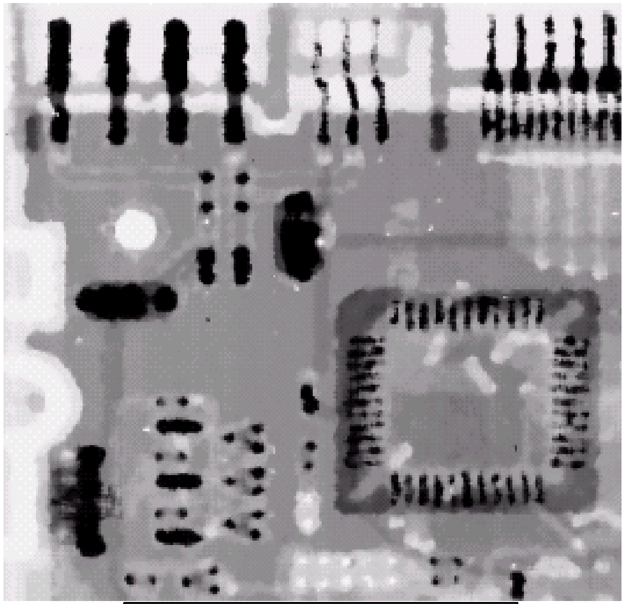


Image obtained using a 7x7 median filter

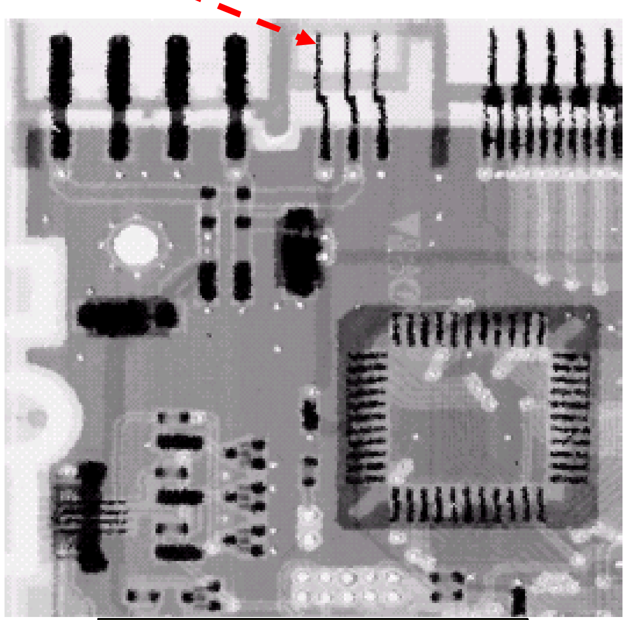


Image obtained using an adaptive median filter with $S_{max}$ = 7

# Thanks for your attention